

The image shows a large, symmetrical pattern composed of black text symbols on a white background. The pattern is roughly triangular in shape, oriented vertically. It features three distinct horizontal bands of symbols. The top band consists of 'SSSSSSSSSSSS' symbols. The middle band consists of 'YYY' symbols. The bottom band consists of 'SSSS' symbols. The arrangement of these symbols creates a visual effect similar to a stylized letter 'A'.

FI. EID**OSWPSCHED

G 1

000000 SSSSSSSS WW WW PBBBBBBB SSSSSSSS CCCCCCCC HH HH EEEEEEEE DDDDDDDD
000000 SSSSSSSS WW WW PBBBBBBB SSSSSSSS CCCCCCCC HH HH EEEEEEEE DDDDDDDD
00 00 SS WW PP PP SS CC HH HH EE DD DD
00 00 SS WW PP PP SS CC HH HH EE DD DD
00 00 SS WW PP PP SS CC HH HH EE DD DD
00 00 SS WW PP PP SS CC HH HH EE DD DD
00 00 SSSSSS WW WW PBBBBBBB SSSSSS CC HHHHHHHHHHHH EEEEEEEE DD DD
00 00 SSSSSS WW WW PBBBBBBB SSSSSS CC HHHHHHHHHHHH EEEEEEEE DD DD
00 00 SS WW PP SS CC HH HH EE DD DD
00 00 SS WW PP SS CC HH HH EE DD DD
00 00 SS WWWWWWWWWWW PP SS CC HH HH EE DD DD
00 00 SS WWWWWWWWWWW PP SS CC HH HH EE DD DD
000000 SSSSSSSS WW WW PP SSSSSSSS CCCCCCCC HH HH EEEEEEEE DDDDDDDD
000000 SSSSSSSS WW WW PP SSSSSSSS CCCCCCCC HH HH EEEEEEEE DDDDDDDD
.....
.....

LL IIIII SSSSSSSS
LL IIIII SSSSSSSS
LL II SS
LLLLLLLLLL IIIII SSSSSSSS
LLLLLLLLLL IIIII SSSSSSSS

(2) 149 DECLARATIONS
(3) 250 SCH\$OSWPSCHED - OUT SWAP SCHEDULER
(4) 487 SHRINK WORKING SET
(5) 544 START MODIFIED PAGE WRITER TEST

0000 1 :TITLE OSWPSCHED - SWAP SCHEDULER
0000 2 :IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28 ++
0000 29 :FACILITY: EXECUTIVE, BALANCE SET SWAPPER
0000 30
0000 31 :ABSTRACT:
0000 32 : OSWPSCHED SELECTS THE LOWEST PRIORITY PROCESS SUITABLE AS AN
0000 33 : OUTSWAP CANDIDATE TO MAKE MEMORY PAGES AVAILABLE BOTH FOR
0000 34 : INSWAPPING AND MAINTENANCE OF THE DESIRED NUMBER OF FREE PAGES.
0000 35
0000 36 :ENVIRONMENT:
0000 37 : MODE = KERNEL, RESIDENT
0000 38
0000 39 :AUTHOR: R. I. HUSTVEDT, CREATION DATE: 14-FEB-77
0000 40
0000 41 :MODIFIED BY:
0000 42
0000 43 :V03-010 SSA0028 Stan Amway 2-Aug-1984
0000 44 : Reverse ordering of LEF and HIB entries in OSWPSTATE.
0000 45
0000 46 :V03-009 WMC0009 Wayne Cardoza 18-Jun-1984
0000 47 : Write modified page list if limit has been set to zero.
0000 48
0000 49 :V03-008 WMC0008 Wayne Cardoza 02-Mar-1984
0000 50 : Check the PHDSM_NO_WS_CHNG flag.
0000 51
0000 52 :V03-007 SSA0014 Stan Amway 5-Mar-1984
0000 53 : Inhibit 2nd level trimming of realtime processes.
0000 54
0000 55 :V03-006 SSA0003 Stan Amway 5-Dec-1983
0000 56 : Reclaim memory loans before doing any 2nd level trimming or
0000 57 : swapping.

0000	58	
0000	59	
0000	60	
0000	61	
0000	62	
0000	63	
0000	64	
0000	65	
0000	66	
0000	67	
0000	68	
0000	69	
0000	70	
0000	71	
0000	72	
0000	73	
0000	74	
0000	75	
0000	76	
0000	77	
0000	78	
0000	79	
0000	80	
0000	81	V03-005 TCM0001 Trudy C. Matthews 31-Mar-1983
0000	82	Change references to working set fields in PHD so that
0000	83	they are used as unsigned words.
0000	84	
0000	85	V03-004 HRJ0210 Herb Jacobs 28-Feb-1982
0000	86	Change priorities to shrink compute processes to quota
0000	87	before outswapping short waiting processes.
0000	88	
0000	89	V03-003 HRJ0100 Herb Jacobs 29-Jun-1982
0000	90	Honor DISAWS in shrinking, and add perturbation into
0000	91	scheduler to recover low priority compute bound processes
0000	92	memory.
0000	93	
0000	94	V03-002 HRJ0061 Herb Jacobs 25-Mar-1982
0000	95	Rewrite again to make it table driven to allow performance
0000	96	measurement of several different scheduling alternatives.
0000	97	Remove V02-008 code, this is accomplished as a side effect
0000	98	of an unsatisfiable pages needed count.
0000	99	
0000	100	V03-001 HRJ0060 Herb Jacobs 20-Mar-1982
0000	101	Add interlock to prevent this code from changing WSLAST
0000	102	out of process context.
0000	103	
0000	104	V02-012 HRJ0052 Herb Jacobs 08-Jan-1982
0000	105	Add sysgen parameter threshold of whether using modified
0000	106	page writer should be used for gaining memory.
0000	107	
0000	108	V02-011 HRJ0051 Herb Jacobs 30-Jan-1982
0000	109	Don't allow a swap candidate if the process is larger than
0000	110	its swap slot. This can happen if free working set list
0000	111	returns a resource wait condition as the result of the
0000	112	preparatory shrink. This should be short lived, and will
0000	113	fix itself on a future need.
0000	114	Also detect a queue reordering as the result of ALLOCPFN

0000 115 : reporting free pages; in this case restart rather than continue.
0000 116 :
0000 117 :
0000 118 : V02-010 HRJ0049 Herb Jacobs 25-Jan-1982
0000 119 : Recompute of extra dynamic working set after MMGSSHRINKWS.
0000 120 :
0000 121 : V02-009 HRJ0044 Herb Jacobs 12-Jan-1982
0000 122 : Update for changed interface to MMGSSHRINKWS and add SWPFAIL
0000 123 : count into priority consideration for outswaps.
0000 124 :
0000 125 : V02-008 HRJ0024 Herb Jacobs 29-Jul-1981
0000 126 : Add back ability to force outswap for shortage of balance set
0000 127 : slots.
0000 128 :
0000 129 : V02-007 HRJ0020 Herb Jacobs 20-Apr-1981
0000 130 : Major enhancement to attempt to shrink processes working sets
0000 131 : to acquire memory before resulting to swapping.
0000 132 :
0000 133 : 06 RIH0059 RICHARD I. HUSTVEDT 25-FEB-1980 19:45
0000 134 : Remove spurious modification of SWTIME.
0000 135 :
0000 136 : 05 RIH0057 RICHARD I. HUSTVEDT 04-FEB-1980 10:07
0000 137 : Remove use of SWPRATE. Moved to SWAPPER instead.
0000 138 :
0000 139 : 04 RIH0036 RICHARD I. HUSTVEDT 01-NOV-1979 11:24
0000 140 : Change wait time measurement to use SCH\$GW_IOTA, a fixed allowance
0000 141 : for voluntary waits to avoid undesired negative performance
0000 142 : feedback.
0000 143 :
0000 144 : 03 RIH0031 RICHARD I. HUSTVEDT 08-AUG-1979 13:17
0000 145 : Remove inhibition on swapping for processes holding a Mutex.
0000 146 :
0000 147 :--

```

149      .SBTTL DECLARATIONS
150
151 : INCLUDE FILES:
152 :
153 :
154     $CEBDEF          ; DEFINE CEB OFFSETS
155     $IPLDEF          ; DEFINE INTERRUPT PRIORITY LEVELS
156     $PCBDEF          ; DEFINE PCB OFFSETS
157     $PHDDEF          ; DEFINE PHD OFFSETS
158     $PRDDEF          ; DEFINE PROCESSOR REGISTERS
159     $STATEDEF         ; DEFINE STATE VALUES
160     $WQHDEF          ; DEFINE WAIT QUEUE HEADER OFFSETS
161
162 : MACROS:
163 :
164 :
165 :
166 :
167     OUTSWP - MACRO TO GENERATE ORDERED ACTION TABLE FOR
168             NON-EXECUTABLE STATES.
169
170     OUTSWP STATE,[<FLAG,FLAG,FLAG...>]
171
172     WHERE: STATE = STATE NAME LESS SCHSC PREFIX
173             FLAG = ONE OF THE SWAP SCHEDULING FLAGS
174                 -V_CEF = COMMON EVENT FLAG STATE
175                 -V_DIOCNT = REQUIRE NONZERO DIOCNT
176                 -V_LONGWAIT = ONLY CONSIDER PROCESSES IN A LONG WAIT
177                 -V_SHORTWAIT = ONLY CONSIDER PROCESSES IN A SHORT WAIT
178                 -V_INQUAN = OBSERVE INITIAL QUANTUM, IGNORE
179                     PROCESS IF PCB$V_INQUAN IS SET
180                 -V_NDIOCNT = IGNORE PROCESS IF PCB$W_DIOCNT IS NONZERO
181                 -V_PRIORITY = OBSERVE PRIORITY OF POSSIBLE OUTSWAP
182                     CANDIDATE RELATIVE TO INSWAP CANDIDATE
183                 -V_COMPUTE = USED TO FLAG START OF COM PROCESS SCAN
184                 -V_SWPOGOAL = REDUCE PROCESS PAST QUOTA BEFORE SWAPPING
185                 -V_SWAPASAP = WHEN A PROCESS IN THIS STATE REACHES A
186                     SWAPPABLE SIZE, EXIT BY SWAPPING IT
187                 -V_DORMANT = ONLY CONSIDER PROCESS IF IT IS DORMANT
188
189     .MACRO OUTSWP,STATE,FLAGS
190     .BYTE SCHSC_,STATE
191 TMP...=0
192     .IRP   FLGS,<FLAGS>
193     TMP...=TMP...+<10_V_>FLGS>
194     .ENDR
195     .WORD  TMP,_
196     .ENDM  OUTSWP
197
198 : EQUATED SYMBOLS:
199 :
200
201     _VIELD  0,<-
202             CEF,-
203             DIOCNT,-
204             LONGWAIT,-
205             SHORTWAIT,-

```

0000	206	INQUAN,-	: OBSERVE INQUAN FLAG
0000	207	NDIOCNT,-	: DIOCNT MUST BE ZERO
0000	208	PRIORITY,-	: OBSERVE PRIORITY
0000	209	COMPUTE,-	: START OF COMPUTE PROCESSES
0000	210	SWPOGOAL,-	: DESIRED SIZE IS SWPOUTPGCNT
0000	211	SWAPASAP,-	: SWAP PROCESSES IN THIS STATE FIFO
0000	212	DORMANT,-	: PROCESS MUST BE DORMANT
0000	213	>	
0000	214	:	
0000	215	OWN STORAGE:	
0000	216	:	
0000	217		
00000000	218	.PSECT \$SS220, LONG	
00000000	219	SWP\$GL_SWTIME::	: EARLIEST TIME FOR NEXT EXCHANGE SWAP
00000000	220	.LONG 0	:
00000000	221	SAVED_R3:	
00000000	222	.LONG	: Used by COMPUTE scan to save state
00000000	223		
00000000	224	.PSECT \$OSWPSCHED, BYTE	
0000	225	OSWPSTATE:	
00 0003	226	OUTSWP SUSP,<SWAPASAP>	: SUSPENDED
00 0004	227	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 0007	228	OUTSWP COM,<DORMANT,COMPUTE,SWAPASAP>	DORMANT PROCESSES
00 0008	229	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 0008	230	OUTSWP HIB,<LONGWAIT,SWPOGOAL>	HIBERNATING, LONG WAIT
00 0008	231	OUTSWP LEF,<NDIOCNT,LONGWAIT,SWPOGOAL>	LOCAL EVENT FLAG WAIT, LONG WAIT
00 000E	232	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 000F	233	OUTSWP CEF,<CEF,NDIOCNT,SWPOGOAL>	COMMON EVENT FLAG WAIT
00 0012	234	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 0013	235	OUTSWP HIB,<SHORTWAIT,SWPOGOAL>	HIBERNATING
00 0016	236	OUTSWP LEF,<NDIOCNT,SHORTWAIT,SWPOGOAL>	LOCAL EVENT FLAG WAIT
00 0019	237	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 001A	238	OUTSWP FPG,<PRIORITY>	FREE PAGE WAIT
00 001D	239	OUTSWP COLPG,<PRIORITY>	COLLIDED PAGE WAIT
00 0020	240	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 0021	241	OUTSWP MWAIT	MUTEX WAIT
00 0024	242	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 0025	243	OUTSWP CEF,<CEF,DIOCNT,INQUAN,PRIORITY>	COMMON EVENT WAIT -DIOCNT NONZERO
00 0028	244	OUTSWP LEF,<DIOCNT,INQUAN,PRIORITY>	LOCAL EVENT WAIT -DIOCNT NONZERO
00 0028	245	.BYTE 0	ATTEMPT OUTSWAPS BEFORE PROCEEDING
00 002C	246	OUTSWP PFW,<INQUAN,PRIORITY>	PAGE FAULT WAIT
00 002F	247	OUTSWP COM,<INQUAN,COMPUTE>	COMPUTE PROCESSES
FF 0032	248	.BYTE -1	END OF TABLE

0033 250 .SBTTL SCH\$OSWPSCHED - OUT SWAP SCHEDULER
 0033 251 ++
 0033 252 : FUNCTIONAL DESCRIPTION:
 0033 253 : SCH\$OSWPCHED SCANS IN LOWEST PRIORITY ORDER FOR PROCESSES THAT
 0033 254 : CAN BE USED TO ACQUIRE MEMORY FROM TO SATISFY THE DEFICIT.
 0033 255 : IF AFTER SHRINKING PROCESSES FAIL, A PROCESS WILL BE SELECTED
 0033 256 : FOR OUTSWAPPING.
 0033 257 :
 0033 258 : CALLING SEQUENCE:
 0033 259 : BSB/JSB SCH\$OSWPSCHED
 0033 260 : IPL = IPL\$_SYNCH
 0033 261 :
 0033 262 : INPUT PARAMETERS:
 0033 263 : SWPSGB ISWPRI - PRIORITY OF INSWAP PROCESS
 0033 264 : FP - NEGATIVE NUMBER OF DESIRED PAGES
 0033 265 :
 0033 266 : IMPLICIT INPUTS:
 0033 267 : ALL STATE QUEUE HEADERS AND ATTACHED PROCESS CONTROL BLOCKS.
 0033 268 :
 0033 269 : OUTPUT PARAMETERS:
 0033 270 : R4 - PCB ADDRESS OF OUTSWAP CANDIDATE
 0033 271 : 0=> NONE
 0033 272 : R5 - PHD ADDRESS OF OUTSWAP CANDIDATE
 0033 273 : R0,R1,R2,R3,R6,R7,R8,R9 DESTROYED
 0033 274 :
 0033 275 : IMPLICIT OUTPUTS:
 0033 276 : SOME PROCESSES WORKING SETS ARE LIKELY TO BE SHRUNK.
 0033 277 : PCB\$V RES OF OUTSWAP CANDIDATE IS CLEARED AND A SWPOUT EVENT
 0033 278 : HAS BEEN REPORTED IF AN OUTSWAP IS TO OCCUR.
 0033 279 :
 0033 280 : COMPLETION CODES:
 0033 281 : NONE
 0033 282 :
 0033 283 : SIDE EFFECTS:
 0033 284 : NONE
 0033 285 :
 0033 286 :--
 0033 287 : .ENABLE LSB
 0033 288 :
 5E 04 C0 0033 289 80\$: ADDL #4,SP : POP CONTINUE ADDRESS OFF STACK
 54 D4 0036 290 90\$: CLRL R4 : GET PROCESS TO SWAP, IF ANY
 05 0038 291 RSB : RETURN TO SWAPPER PROPER
 0039 292 :
 0039 293 : RETRY:
 SE 04 C0 0039 294 ADDL #4,SP : USED IN CASE ALLOCPFN REORDERS QUEUES
 003C 295 : POP CONTINUE ADDRESS FROM STACK AND
 0232 30 003C 296 SCH\$OSWPSCHED:: TRY AGAIN
 5D D5 003F 297 BSBW MMGSMPWCHECK : SCHEDULE OUTSWAP
 F3 18 0041 298 TSTL FP : WILL STARTING MPW SATISFY NEED?
 0000'CF 94 0043 299 BGEQ 90\$: FP CONTAINS DEFICIT
 0047 300 CLRBL W^SCH\$GB_RESCAN : BRANCH IF NO MORE DEFICIT
 0047 301 : CLEAR RESCAN NEEDED FLAG
 0047 302 : Begin 1st level trimming
 0047 303 :
 53 B6 AF 9E 0047 304 MOVAB OSWPSTATE,R3 : INITIALIZE STATE SCAN AT TOP
 52 53 D0 004B 305 10\$: MOVL R3,R2 : RESET TO NEW SUB SECTION
 28 13 004E 306 BEQL 24\$: BRANCH IF AT END OF TABLE

00F6 30 0050 307 BSBW CANDIDATE : GET A CANDIDATE TO BE SHRUNK
 F6 13 0053 308 BEQL 10\$: BRANCH IF NONE, TRY NEXT SUB SECTION
 58 08 A5 A3 0055 309 20\$: SUBW3 PHDSW_WSLIST(R5),PHDSW_WSQUOTA(R5),R8
 58 B6 005B 310 INCW R8 : SIZE TO SHRINK WORKING SET TO
 01D9 30 005D 311 BSBW SHRINKWS : RECLAIM MEMORY FROM IT
 D1 0000'CF 00' D1 18 0060 312 BGEQ 80\$: IF NO DEFICIT, QUIT TRIMMING
 06 56 09 E0 0062 313 BBC S^#SCH\$V REORD,W^SCH\$GB_RESCAN,RETRY :BRANCH IF FPG STATE CHANGE
 006C 30 006C 314 BSBW # V SWAPASAP,R6,22\$: IF A FULL PROCESS SWAP IS DESIRED
 53 59 E8 006F 315 BLBS R9,60\$: AND IT IS CURRENTLY POSSIBLE,
 9E 16 0072 317 22\$: JSB @(\$P)+ : DO IT NOW
 DF 12 0074 318 BNEQ 20\$: TRY FOR THE NEXT ONE
 D3 11 0076 319 BRB 10\$: BRANCH IF THERE IS ONE
 0078 320 :
 0078 321 : Begin 2nd level trimming and swapping
 0078 322 :
 53 85 AF 9E 0078 323 24\$: MOVAB OSWPSTATE,R3 : RESTART STATE SCAN AT TOP
 52 53 D0 007C 324 25\$: MOVL R3,R2 : RESET TO NEW SUB SECTION
 85 13 007F 325 BEQL 90\$: BRANCH IF AT END OF TABLE
 00C5 30 0081 326 BSBW CANDIDATE : GET A CANDIDATE TO BE SHRUNK
 F6 13 0084 327 BEQL 25\$: BRANCH IF NONE, TRY NEXT SUB SECTION
 0161 30 0086 328 30\$: BSBW SWAPQUANCHK : CHECK SWAPABILITY AND INITIAL QUANTUM
 27 59 E9 0089 329 BLBC R9,40\$: BRANCH IF SHOULD NOT SWAP IT NOW
 OF 24 A4 18 E0 008C 330 BBS #PCBV_DISAWS,PCBSL_STS(R4),32\$;BRANCH IF CAN'T SHRINK IT
 0B A4 10 91 0091 331 CMPB #16,PCBSB_PRI(R4) : Real-time process ?
 09 1A 0095 332 BGTRU 32\$: BR if yes - can't shrink RT process
 58 0000'CF D0 0097 333 MOVL W^SWPSGL_SWPPGCNT,R8 : VALUE TO SHRINK CANDIDATE TO
 08 56 08 E0 009C 334 BBS # V SWPOGOAL,R6,35\$: BRANCH IF THIS IS DESIRED GOAL SIZE
 58 18 A5 08 A5 A3 00A0 335 32\$: SUBW3 PHDSW_WSLIST(R5),PHDSW_WSQUOTA(R5),R8
 58 B6 00A6 336 INCW R8 : SIZE TO SHRINK WORKING SET TO
 018E 30 00A8 337 35\$: BSBW SHRINKWS : RECLAIM MEMORY FROM IT
 86 0000'CF 00' E0 00AD 339 BBS S^#SCH\$V_REORD,W^SCH\$GB_RESCAN,RETRY ; BRANCH IF FPG STATE CHANGE
 9E 16 00B3 340 40\$: JSB @(\$P)+ : TRY FOR NEXT ONE
 CF 12 00B5 341 BNEQ 30\$: BRANCH IF THERE IS ONE
 008F 30 00B7 342 BSBW CANDIDATE : GET A CANDIDATE TO BE OUTSWAPPED
 CO 13 00BA 343 50\$: BEQL 25\$: NO ONE TO OUTSWAP
 1D 10 00BC 344 BSBB 100\$: CHECK SWAPABILITY AND INITIAL QUANTUM
 04 59 E8 00BE 345 BLBS R9,60\$: BRANCH TO SWAP IT
 9E 16 00C1 346 JSB @(\$P)+ : TRY FOR NEXT ONE
 F5 11 00C3 347 BRB 50\$: IS THERE ANOTHER ONE?
 0000'CF 0000'CF 52 D4 00C5 349 60\$: CLRL R2 : NULL PRIORITY INCREMENT
 0000'CF 0000'CF 00 00C7 350 BBCCI #PCBV_RES,PCBSL_STS(R4),70\$; CLEAR RESIDENT FLAG
 SE 04 B0 00D0 351 70\$: RPTEVT SWPOUT : REPORT SWPOUT EVENT
 CO 00D7 352 MOVW W^SCH\$GW_SWPFFAIL,W^SCH\$GW_SWPFCNT : RESET FAILURE COUNTER
 05 00DA 353 ADDL #4,SP : POP CONTINUE ADDRESS OFF STACK
 00DB 354 RSB : WITH PCB IN R4, PHD IN R5
 59 34 A4 36 A4 A1 00DB 356 100\$: ADDW3 PCBSW_PPGCNT(R4),PCBSW_GPGCNT(R4),R9 ;GET SIZE OF PROCESS
 52 A5 59 B1 00E1 357 CMPW R9 PHDSW_SWAPSIZE(R5) : DO WE HAVE ENOUGH SPACE TO SWAP IT?
 03 1B 00E5 358 BLEQU 110\$: BRANCH IF YES
 59 D4 00E7 359 CLRL R9 : SET FAILURE
 05 00E9 360 RSB : RETURN
 00FD 30 00EA 361 110\$: BSBW SWAPQUANCHK : CHECK SWAPABILITY AND INITIAL QUANTUM
 00ED 05 00EE 362 RSB
 00EE 363

06 56 0A 00EE 364 .DISABLE LSB
 0000'CF 00EE 365 .ENABLE LSB
 0004'CF 00EE 366 COMPUTE:
 51 54 13 E1 00F2 367 BBC # V DORMANT,R6,5\$
 0000'CF 53 00FD 368 TSTW W\$ SCHSGW DORMANTWAIT : COMPUTE STATE SCAN
 53 71 13 00F6 369 BEQL NEXTSTATE : Prevent DORMANT process scan
 0000'CF 53 00F8 370 5\$: MOVL R3,W^SAVED,R3 : if DORMANTWAIT = 0
 51 54 13 0102 371 MOVAB W^SCHSAQ_COMOH,R1
 2E 7D 13 0105 372 10\$: BEQL -(R1),R3 : Save state for scan continuation
 54 51 0107 373 20\$: MOVQ 50\$: POINT TO LONGWORD BEYOND PRI=0 HDR
 07 56 0A F6 13 010A 374 BEQL 50\$: GET QUEUE HEADER
 00F7 010C 375 CMPL R1,R4 : NONE, SCAN FINISHED
 2A 30 0110 376 BBC # V DORMANT,R6,30\$: CHECK FOR EMPTY QUEUE
 18 19 0113 377 BSBW DORMANT_CHECK : YES, NEXT QUEUE (HIGHER PRIORITY)
 18 13 0115 378 BLSS 55\$: Must process be DORMANT ?
 0000'CF 0B A4 0117 379 BEQL 40\$: Process must be DORMANT - Is it ?
 16 19 011D 380 CMPB PCB\$B_PRI(R4),W^SWP\$GB_ISWPRI : If LSS, terminate scan
 55 6C A4 011F 381 BLSS 50\$: BRANCH IF CAN'T SWAP
 0123 382 MOVL PCB\$L_PHD(R4),R5 : MMG ROUTINES WANT PHD IN R5
 36 A5 40 8F 93 0123 383 ASSUME PHDSM_NO WS CHNG LE 255
 05 12 0128 384 BITB #PHDSM_NO_WS_CHNG,PHD\$W_FLAGS(R5) : ARE WE ALLOWED TO TOUCH IT
 55 55 012A 385 BNEQ 40\$: NO
 9E 16 012D 386 MOVL R5,R5 : SET Z=0
 54 04 A4 012F 387 JSB @(\$P)+ : RETURN WITH CANDIDATE, WE MAY CONTINUE
 D2 11 0133 388 MOVL PCB\$L_SQBL(R4),R4 : BLINK THROUGH QUEUE
 0135 389 40\$: BRB 20\$: AND CHECK FOR EMPTY
 0000'CF B7 0135 390 DECW W\$ SCHSGW_SWPF CNT : COUNT A FAILURE
 04 18 0139 391 BGEQ 55\$: EXIT IF NO UNDERFLOW
 53 0004'CF B4 013B 392 CLRW W\$ SCHSGW_SWPF CNT : LIMIT VALUE TO ZERO
 0000'CF D0 013F 393 55\$: MOVL W^SAVED,R3,R3 : Restore state and continue
 06 11 0144 394 BRB NEXTSTATE
 0146 395 55\$: INDICATE NONE FOUND, Z BIT SET, END
 53 D4 0146 396 R3 : AND RETURN
 05 0148 397 400
 0149 401 CANDIDATE:
 53 52 D0 0149 402 MOVL R2,R3 : INIT STATE SCAN AT CURRENT SUB SECTION
 014C 403 NEXTSTATE: TRY NEXT STATE
 014C 404 CLRL R1 : (MOVZBL not used so that the two
 51 51 D4 014C 405 conditional tests will work correctly)
 83 90 014E 406 MOVB (R3)+,R1 : GET NEXT STATE TO TRY
 F5 13 0151 407 BEQL 70\$: BRANCH IF AT END OF STATE SUB SECTION
 F1 19 0153 408 BLSS 60\$: Branch if at end of table
 56 83 3C 0155 409 MOVZWL (R3)+,R6 : FETCH STATE FLAGS WORD
 92 56 07 E0 0158 410 BBC # V COMPUTE,R6,COMPUTE : TRY COM IF CANT FIND NON-EXECUTABLE
 1A 56 00 E1 015C 411 BBC # V CEF,R6,110\$: CONTINUE IF NOT CEF WAIT
 57 0000'CF 7E 0160 412 MOVAQ W\$ SCHSGQ_CEBHD,R7 : GET ADDRESS OF LISTHEAD
 50 14 A7 DE 0165 413 MOVAL CEB\$L_WQFL(R7),R0 : AND MAKE WORKING COPY
 0169 414 NEXTCEB: NEXT CEB
 50 E0 56 E9 0169 415 BLBC R6,NEXTSTATE : BR IF V CEF CLEAR
 EC A0 D0 016C 416 MOVL -CEB\$L_WQFL(R0),R0 : FLINK FORWARD
 57 50 D1 0170 417 CMPL R0,R7 : CHECK FOR END OF CEB LIST
 D7 13 0173 418 BEQL NEXTSTATE : AT END IF EQL
 50 14 C0 0175 419 ADDL #CEB\$L_WQFL,R0 : POINT TO WAIT QUEUE
 OA 11 0178 420 BRB 120\$: ELSE SCAN CEB WAIT QUEUE

50 0000'CF41 7E 017A 421 110\$: MOVAQ W\$SCH\$AQ_WQHDR[R1],R0 ; COMPUTE ADDRESS OF WAIT QUEUE HDR
 50 6041 DE 0180 422 MOVAL (R0)[R1],R0 ; AS SCH\$AQ_WQHDR+12*R1
 54 04 A0 DO 0184 423 120\$: MOVL WQHSL_WQBL(R0),R4 ; GET TAIL OF QUEUE
 0188 424
 0188 425 .DISABLE LSB
 0188 426
 50 54 D1 0188 427 WSCAN: CMPL R4,R0 ; CHECK FOR END OF QUEUE
 DC 13 0188 428 BEQL NEXTCEB ; TRY NEXT STATE OR CEB
 53 24 A4 E9 018D 429 BLBC PCB\$L_STS(R4),NEXTPR ; NOT RESIDENT, NEXT IN QUEUE
 40 A4 3E A4 B1 0191 430 CMPW PCB\$W_DIOCNT(R4),PCBSW_DIOLM(R4) ; CHECK IT FOR ZERO
 06 13 0196 431 BEQL 10\$; IF ZERO, IS THAT PERMITTED?
 48 56 05 E0 0198 432 BBS # V_NDIOCNT,R6,NEXTPR ; BR IF NON-ZERO NOT PERMITTED
 04 11 019C 433 BRB 20\$; CONTINUE IF NON-ZERO ALLOWED
 42 56 01 E0 019E 434 10\$: BBS # V_DIOCNT,R6,NEXTPR ; BR IF ZERO DIOCNT NOT PERMITTED
 0E 56 06 E1 01A2 435 20\$: BBC # V_PRIORITY,R6,30\$; BR IF PRIORITY CAN BE IGNORED
 0000'CF 08 A4 91 01A6 436 CMPB PCBSB_PRI(R4),W\$SWPSGB_ISWPRI ; COMPARE WITH INSWAP PRIORITY
 06 18 01AC 437 BGEQ 30\$; BRANCH IF LESS IMPORTANT THAN ISWP
 0000'CF B5 01AE 438 TSTW W\$SCH\$GW_SWPFcnt ; CHECK FOR FAILURE COUNT EXHAUSTED
 30 14 01B2 439 BGTR NEXTPR ; BRANCH IF NOT GOOD CHOICE YET
 56 0C 93 01B4 440 30\$: BITB #<10_V_SHORTWAIT>!<10_V_LONGWAIT>,R6 ; CHECK WAIT TIME?
 18 13 01B7 441 BEQL 50\$; BRANCH IF NOT
 55 0000'CF 3C 01B9 442 MOVZWL W\$SCH\$GW_LONGWAIT,R5 ; Get long wait threshold
 55 0118 C4 C0 01BE 443 ADDL2 PCB\$L_WAITIME(R4),R5 ; R5 <= wait time + delta
 55 0000'CF D1 01C3 444 CMPL W\$EXE\$GL_ABSTIM,R5 ; HOW LONG HAVE WE BEEN WAITING?
 02 56 03 E1 01C8 445 BBC # V_SHORTWAIT,R6,40\$; BRANCH IF NON- SHORT WAIT ALLOWED
 16 1A 01CC 446 BGTR NEXTPR ; BRANCH IF WAIT INTERVAL IS LONG
 02 56 02 E1 01CE 447 40\$: BBC # V_LONGWAIT,R6,50\$; BRANCH IF NON- LONG WAIT ALLOWED
 10 1F 01D2 448 BLSSU NEXTPR ; BRANCH IF WAIT INTERVAL IS SHORT
 55 6C A4 DO 01D4 449 50\$: MOVL PCBSL_PHD(R4),R5 ; MMG ROUTINES WANT PHD IN R5, SET Z=0
 01D8 450 ASSUME PHDSM_NO_WS_CHNG LE 255
 36 A5 40 8F 93 01D8 451 BITB #PHDSM_NO_WS_CHNG,PHDSW_FLAGS(R5) ; ARE WE ALLOWED TO TOUCH IT
 05 12 01DD 452 BNEQ NEXTPR ; NO
 55 55 D0 01DF 453 MOVL R5,R5 ; SET Z=0
 9E 16 01E2 454 JSB @(SP)+ ; RETURN SO WE CAN CONTINUE
 54 04 A4 DO 01E4 455 NEXTPR: MOVL PCBSL_SQBL(R4),R4 ; BLINK THROUGH QUEUE
 9E 11 01E8 456 BRB WSCAN ; AND CONTINUE SCAN
 01EA 457
 01EA 458 SWAPQUANCHK: ; CHECK VALIDITY OF PROCESS CHOICE
 59 D4 01EA 459 CLRL R9 ; ASSUME BAD CHOICE
 18 24 A4 04 E0 01EC 460 BBS #PCBSV_PSWAPM,PCBSL_STS(R4),20\$; BRANCH IF NOT SWAPPABLE
 12 56 04 E1 01F1 461 BBC # V_INQUAN,R6,10\$; BRANCH IF INQUAN CAN BE IGNORED
 0D 24 A4 03 E1 01F5 462 BBC #PCBSV_INQUAN,PCBSL_STS(R4),10\$; BRANCH IF INQUAN SATISFIED
 OF 0000'CF 91 01FA 463 CMPB W\$SWPSGB_ISWPRI,#15 ; IS THIS A REAL TIME INSWAP?
 06 15 01FF 464 BLEQ 10\$; BRANCH IF SO
 0000'CF B5 0201 465 TSTW W\$SCH\$GW_SWPFcnt ; CHECK FOR FAILURE COUNT EXHAUSTED
 02 14 0205 466 BGTR 20\$; BRANCH IF NOT GOOD CHOICE YET
 59 D6 0207 467 10\$: INCL R9 ; SET GOOD CHOICE
 05 0209 468 20\$: RSB ; RETURN WITH STATUS IN R9
 020A 469
 020A 470 DORMANT_CHECK: ; Check if candidate is DORMANT
 7E 01 CE 020A 471 MNEG L #1,-(SP) ; Preset terminate scan status
 55 0B A4 90 020D 472 MOVB PCBSB_PRI(R4),R5 ; Fetch priority of candidate
 55 10 91 0211 473 CMPB #16,R5 ; Real-time process ?
 20 1A 0214 474 BGTR 10\$; BR if yes
 58 0000'CF 83 0216 475 SUBB3 W\$SYSSGB_DEFPRI,#31,R8 ; Convert to internal form for test
 58 55 91 021C 476 CMPB R5,R8 ; Is this a low-priority job ?
 15 1F 021F 477 BLSSU 10\$; BR if not

58	0000'CF	6E	D6	0221	478	INCL	(SP)		: Set bad candidate, continue status
58	0118 C4	3C	0223	479	MOVZWL	W\$SCH\$GW_DORMANTWAIT,R8		: Get dormant wait threshold	
58	0000'CF	C0	0228	480	ADDL2	PCB\$L_WAITIME(R4),R8		: Has process had a significant	
		D1	022D	481	CMPL	W\$EXESGL_ABSTIM,R8		: event recently ?	
		02	1F	0232	482	BLSSU	10\$: BR if not	
		6E	D6	0234	483	INCL	(SP)	: Indicate that process is DORMANT	
		8E	D5	0236	484	TSTL	(SP)+	: Set status in condition codes	
		05	0238	485	RSB				

10\$:

0239 487 .SBTTL SHRINK WORKING SET
 0239 488 :++
 0239 489 : FUNCTIONAL DESCRIPTION:
 0239 490 SHRINK IS USED TO MAKE PAGES AVAILABLE FROM OTHER PROCESSES THAT
 0239 491 HAVE BORROWED PAGES TO SATISFY HIGH FAULTING RATES OR TO SHRINK
 0239 492 A SWAPPABLE PROCESS TO THE DESIRED SIZE FOR SMALLER SWAPS.
 0239 493 : CALLING SEQUENCE:
 0239 494 BSB/JSB SHRINKWS
 0239 495 : INPUT PARAMETERS:
 0239 496 R4 - PCB OF TARGET PROCESS
 0239 497 R5 - PHD OF TARGET PROCESS
 0239 498 R8 - SIZE TO ATTEMPT TO SHRINK PROCESS TO
 0239 499 FP - NEGATIVE NUMBER OF TOTAL PAGES WANTED BY SYSTEM
 0239 500 : IMPLICIT INPUTS:
 0239 501 WORKING SET LIST
 0239 502 : OUTPUT PARAMETERS:
 0239 503 FP UPDATED
 0239 504 R9 DESTROYED
 0239 505 : IMPLICIT OUTPUTS:
 0239 506 TARGET PROCESS WORKING SET SHRUNK UP TO R8 INPUT VALUE
 0239 507 : COMPLETION CODES:
 0239 508 POSITIVE INDICATES SWAPPER REQUEST SATISFIED
 0239 509 : SIDE EFFECTS:
 0239 510 THIS ROUTINE COULD POTENTIALLY DUMP A LOT OF TRAFFIC INTO THE
 0239 511 PAGING SYSTEM, ESPECIALLY THE MODIFIED PAGE WRITER.

59 58 58 3C 0239 523 MOVZWL R8,R8 : GET INPUT SIZE IN A LONGWORD
 59 50 A5 3C 023C 524 MOVZWL PHDSW_WSSIZE(R5),R9 : GET CURRENT SIZE
 59 58 C3 0240 525 SUBL3 R9 R8,R9 : CAN WE GET ANY PAGES FROM PROCESS?
 28 18 0244 526 BGEQ 30\$: BRANCH IF NOT
 OF BB 0246 527 PUSHR #^M<R0,R1,R2,R3> : SAVE SWAPPER OUTSWAP SCHEDULER REGS
 51 59 DD 0248 528 MOVL R9,R1 : NUMBER OF PAGES TO SHRINK PROCESS BY
 0000'CF 0000'CF 00 00 0248 529 PUSHL W^\$CH\$GL_FREECNT : SAVE OLD FREE LIST COUNT
 0000'CF 00 00 024F 530 BISB S^#MMGSM_NOLASTUPD!MMGSM NOWAIT,W^MMG\$GB FREWFLGS : DISABLE
 0000'CF 0000'CF 00 00 0254 531 BSBW MMGSSHINKWS : WSLAST UPDATE AND MP LIST SIZE WAIT
 0000'CF 00 00 0254 532 BICB S^#MMGSM_NOLASTUPD!MMGSM NOWAIT,W^MMG\$GB FREWFLGS : USE ADJWSL CODE, IPL REMAINS AT SYNCH
 0000'CF 00 00 0257 533 BSBW MMGSEXTRADYNWS : REENABLE
 0000'CF 00 00 025C 534 BICB S^#MMGSM_NOLASTUPD!MMGSM NOWAIT,W^MMG\$GB FREWFLGS : WSLAST UPDATE AND MP LIST SIZE WAIT
 59 F0 0000'CF 0000'CF 00 00 025C 535 BSBW (SP)+,W^\$CH\$GL_FREECNT,R9 : RECALC EXTRA DYNAMIC WORKING SET
 5D 8E C3 025F 536 SUBL3 ADDL R9,FP : CALC NUMBER OF PAGES FREED
 5D 59 C0 0265 537 BGEQ 20\$: HAVE WE SATISFIED DEFICIT?
 02 18 0268 538 TSTL FP : BRANCH IF YES
 05 10 026A 539 BSBB MMGSMPWCHECK : WILL STARTING MPW SATISFY NEED?
 0F BA 026C 540 20\$: POPR #^M<R0,R1,R2,R3> : RESTORE OSWPSCHED REGISTERS
 5D D5 026E 541 30\$: TSTL FP : SET DEFICIT INDICATION
 05 0270 542 RSB : RETURN

0271 544 .SBTTL START MODIFIED PAGE WRITER TEST
 0271 545 :++
 0271 546 : FUNCTIONAL DESCRIPTION:
 0271 547 : THIS ROUTINE IS CALLED BY THE SWAPPER TO OPTIONALLY START THE
 0271 548 : MODIFIED PAGE WRITER IF IT WILL SOLVE THE CURRENT DEFICIT PROBLEM.

0271 549 : CALLING SEQUENCE:
 0271 550 : BSB/JSB MMGSMPWCHECK
 0271 551 :
 0271 552 :
 0271 553 : INPUT PARAMETERS:
 0271 554 : FP - NEGATIVE NUMBER OF TOTAL PAGES WANTED BY SYSTEM

0271 555 :
 0271 556 : IMPLICIT INPUTS:
 0271 557 : MODIFIED PAGE WRITER VALUES

0271 558 :
 0271 559 : OUTPUT PARAMETERS:
 0271 560 : FP UPDATED TO REFLECT HOW MUCH IS EXPECTED BY RUNNING MPW

0271 561 :
 0271 562 : IMPLICIT OUTPUTS:
 0271 563 : MODIFY PAGE WRITER LOW THRESHOLD MAY BE SET TO START MPW

0271 564 :
 0271 565 : COMPLETION CODES:
 0271 566 : NONE

0271 567 :
 0271 568 : SIDE EFFECTS:
 0271 569 : NONE

0271 570 :
 0271 571 :--
 0271 572 :
 0271 573 : MMGSMPWCHECK::

0000'CF		B5	0271	574	TSTW	W^SCH\$GL_MFYLIM	: HAS LIMIT BEEN SET TO ZERO - LOW WORD
OF	12	0275	575	BNEQ	10\$: NO CONTINUE WITH NORMAL CHECKS	
0000'CF	D5	0277	576	TSTL	W^SCH\$GL_MFYCNT	: MAKE SURE SOMETHING IS REALLY THERE	
2B	13	027B	577	BEQL	30\$: NOTHING TO DO	
50	SD 01	78	027D	578	ASHL	#1 FP, R0	: GET RID OF HIGH BIT
	03	12	0281	579	BNEQ	10\$: IT WASN'T THE "FREE BALANCE SLOT" FLAG
	5D	D4	0283	580	CLRL	FP	: SIGNAL THE START OF THE MPW
		05	0285	581	RSB		
0000'CF	0000'CF	D1	0286	582	CMPL	W^MPWSGL_THRESH, W^SCH\$GL_MFYCNT;	: IS MPW VALID FOR GETTING MEMORY
	19	14	028D	583	BGTR	30\$: BRANCH IF NOT
50	0000'CF	C3	028F	584	SUBL3	W^SCH\$GL_MFYLOLIM, W^SCH\$GL_MFYCNT, R0;	: WILL WRITING MODIFIED
	OF	15	0297	585	BLEQ	30\$: PAGES HELP?, BR IF NONE
	50	SD C0	0299	586	ADDL	FP, R0	: ARE THERE ENOUGH?
0000'CF	0000'CF	OA	029C	587	BLSS	30\$: BR IF NOT AND AVOID FLUSHING LIST
	5D	50	029E	588	MOVW	W^SCH\$GL_MFYLOLIM, W^SCH\$GL_MFYLM;	: RESET LOWER THRESHOLD
		00	02A5	589	MOVL	R0, FP	: INDICATE NEW DEFICIT
		05	02A8	590	50\$:	RSB	: EXIT, MPW WILL START IF NECESSARY
			02A9	591			
			02A9	592		.END	

OSWPSCHED Symbol table

- SWAP SCHEDULER

H 2

16-SEP-1984 00:40:12 VAX/VMS Macro V04-00
5-SEP-1984 03:45:41 [SYS.SRC]OSWPSCHED.MAR;1

Page 13
(5)

SY
VO

BIT
CANDIDATE
CEBSL_WQFL
COMPUTE
DORMANT_CHECK
EVTS_SWPOUT
EXESGL_ABSTIM
MMGSEXTRADYNWS
MMGSGB_FREWFLLGS
MMGSMPQCHECK
MMGSM_NOLASTUPD
MMGSM_NOWAIT
MMGSSRINKWS
MPWSDL_THRESH
NEXTCEB
NEXTPR
NEXTSTATE
OSWPSTATE
PCBSB_PRI
PCBSL_PHD
PCBSL_SQBL
PCBSI_STS
PCBSL_WAITIME
PCBSV_DISAWS
PCBSV_INQUAN
PCBSV_PSWAPM
PCBSV_RES
PCBSW_DIOCNT
PCBSW_DIOLM
PCBSW_GPGCNT
PCBSW_PPGCNT
PHDSM_NO_WS_CHNG
PHDSW_FLAGS
PHDSW_SWAPSIZE
PHDSW_WSLIST
PHDSW_WSQUOTA
PHDSW_WSSIZE
RETRY
SAVED_R3
SCHSAQ_COMOH
SCHSAQ_WQHDR
SCHSC_CEF
SCHSC_COLPG
SCHSC_COM
SCHSC_FPG
SCHSC_HIB
SCHSC_LEF
SCHSC_MWAIT
SCHSC_PFW
SCHSC_SUSP
SCHSGB_RESCAN
SCHSGL_FREECNT
SCHSGL_MFYCNT
SCHSGL_MFYLIM
SCHSGL_MFYLDLIM
SCHSGQ_CEBHD
SCHSGW_DORMANTWA

```
+-----+
! Psect synopsis !
+-----+
```

PSECT name

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$220	00000008 (8.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
SOSWPSCHED	000002A9 (681.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

```
+-----+
! Performance indicators !
+-----+
```

Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.04	00:00:01.24
Command processing	120	00:00:00.48	00:00:04.34
Pass 1	216	00:00:05.92	00:00:22.00
Symbol table sort	0	00:00:00.71	00:00:01.73
Pass 2	123	00:00:01.70	00:00:04.64
Symbol table output	11	00:00:00.08	00:00:00.08
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	505	00:00:08.96	00:00:34.07

The working set limit was 1200 pages.

33660 bytes (66 pages) of virtual memory were used to buffer the intermediate code.
 There were 30 pages of symbol table space allocated to hold 456 non-local and 39 local symbols.
 592 source lines were read in Pass 1, producing 19 object records in Pass 2.
 18 pages of virtual memory were used to define 17 macros.

```
+-----+
! Macro library statistics !
+-----+
```

Macro library name

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB:1	7
\$255\$DUA28:[SYSLIB]STARLET.MLB:2	6
TOTALS (all libraries)	13

505 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:OSWPSCHED/OBJ=OBJ\$:OSWPSCHED MSRC\$:OSWPSCHED/UPDATE=(ENH\$:OSWPSCHED)+EXECMLS/LIB

0378 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PAGEFAULT
LTS

OSWPSCHED
LTS

PISYSVECT
LTS

PARAMETER
LTS

PAGEFILE
LTS